

## **Ansätze zur Reduktion der Live-Migrationsdauer virtueller Maschinen**

Evgeny Semenov, Daniel Versick, Djamshid Tavangarian

Forschungsgruppe Rechnerarchitektur  
Institut für Informatik, Universität Rostock  
Joachim-Jungius-Str. 9  
18055 Rostock  
vorname.nachname@uni-rostock.de

**Abstract:** Viele Rechenzentren nutzen Virtualisierungstechnologien zur Verbesserung der Auslastung vorhandener Server-Ressourcen. Um in derart virtualisierten Umgebungen die Wartung der physikalischen Systeme zu realisieren bzw. Ausfälle adäquat behandeln zu können, werden vom Administrator gesteuerte manuelle Migrationen der virtuellen Maschinen (VM) eingesetzt. Automatische Live-Migrationen, die nicht vom Administrator sondern vom Virtualisierungssystem eigenständig bei einer Änderung von Anforderungen gestartet werden, könnten eine maßgebliche Bedeutung für eine Vielzahl von zusätzlichen Anwendungsfällen in Rechenzentren der Zukunft haben. Zur Realisierung einer automatischen Live-Migration ist die Reduktion der Migrationszeiten und insbesondere die Minimierung der Dauer der Nicht-Erreichbarkeit einer virtuellen Maschine (Offline-Zeit) entscheidend, da bei automatischen Verfahren kein Eingriff des Administrators in den Migrationsvorgang erfolgen sollte. In dem vorliegenden Beitrag werden Bedingungen analysiert, die zu hohen Migrations- und Offline-Zeiten bei automatisierten Live-Migrationen führen können. Dazu werden experimentelle Messungen und ihre Testergebnisse vorgestellt, um Abhängigkeiten der Migrations- und Offline-Zeiten bei unterschiedlichen Prozessor- und Speicherlastsituationen in Xen-basierten Umgebungen zu zeigen. Es werden Testergebnisse präsentiert und deren Einsatz in automatischen Live-Migrationsumgebungen diskutiert. Unter Nutzung dieser Ergebnisse können hohe Migrations- und Offline-Zeiten vermieden werden, die sonst zu einer Verringerung der Service-Qualität und im schlimmsten Fall zu Dienstaussfällen führen können.

**Keywords:** Server-Virtualisierung, Live-Migration, Xen, Performance, Speicherlast, Prozessorlast.

## 1. Virtualisierung in Rechenzentren

Virtualisierungslösungen werden seit Jahren in Rechenzentren eingesetzt, um die Auslastung der physikalischen Server-Ressourcen zu verbessern und ihre Administration zu vereinfachen. Aus diesem Grund ist Virtualisierung zu einem wichtigen Motor zur Senkung der Betriebskosten geworden [1]. In heutigen Rechenzentren werden Virtualisierungstechniken in zahlreichen Komponenten eingesetzt, unter anderem Server, Netzwerk und Speicher [2], wobei sowohl Aggregationsverfahren zur Kombination vieler physikalischer Ressourcen zu einer virtuellen Ressource als auch Partitionierungsverfahren zur Aufteilung physikalischer in virtuelle Ressourcen eingesetzt werden.

Im vorliegenden Beitrag wird das Thema Server-Virtualisierung fokussiert – eine Partitionierungstechnologie, bei der die Ressourcen eines physikalischen Serversystems durch mehrere virtuelle Maschinen (VMs) mit unterschiedlichen Betriebssystemen und Anwendungssoftware genutzt werden. Allerdings hat dieser Vorteil der Virtualisierung – also, die hohe Auslastung der Serverhardware, indem mehrere virtuelle Maschinen auf wenigen physikalischen Hosts laufen – auch einen wesentlichen Nachteil [3]: Technische Probleme auf einem physikalischen Host können Service-Ausfälle aller VMs auf diesem Host zur Folge haben [4]. In modernen Rechenzentren werden Migrationstechnologien eingesetzt, um diese Probleme durch ein Verschieben der VMs auf andere physikalische Systeme vor dem Auftreten der technischen Probleme zu vermeiden [5]. Da durch Virtualisierung für das Betriebssystem eine Abstraktion von der Hardware stattfindet, ist es relativ einfach, eine virtuelle Maschine von einem physikalischen Serversystem (*Quellhost*) auf ein anderes physikalisches System (*Zielhost*) zu migrieren.

### 1.1. Konzepte der VM-Migration

Es gibt vier verschiedene Ansätze zur Durchführung der Migration virtueller Maschinen, wobei jeder der Ansätze aus einer oder mehreren der im Folgenden aufgeführten Migrationsphasen besteht [6]:

1. *Push-Phase*: Die Speicherseiten der VM werden vom Quellhost zum Zielhost gesendet, während die VM weiter auf dem Quellhost ausgeführt wird; die im Laufe eines Kopiervorgangs geänderten Seiten müssen in weiteren Kopier-Iterationen erneut gesendet werden, wobei der Vorgang nach einer implementierungsabhängigen Iterationsanzahl abgebrochen wird;
2. *Stop-and-Copy-Phase*: Die Speicherseiten und VM-Zustand werden nach dem Anhalten der zu migrierenden VM übertragen; danach wird die VM auf dem Zielhost gestartet;
3. *Pull-Phase*: Nicht übertragene Speicherseiten werden während der Laufzeit der VM auf dem Zielhost auf diesen Host übertragen, sobald die VM auf diese Seiten zugreift.

Der einfachste und gleichzeitig klassische Ansatz zur Migration virtueller Maschinen ist die Stop-and-Copy-Migration, die nur aus der zweiten Phase besteht und vor allem einen großen Nachteil hat: Während des gesamten Kopiervorganges muss die VM gestoppt werden, um einen inkonsistenten Zustand und Datenverlust zu vermeiden. Innerhalb dieser Zeitspanne ist eine Erreichbarkeit der VM nicht gewährleistet (Offline-Zeit), so dass es in einem Rechenzentrum zu Beeinträchtigungen der Dienstverfügbarkeit kommt. Bei einer VM mit hohem Speicherumfang kann die Offline-Zeit während der VM-Migration inakzeptabel lang werden [6].

Um die VM-Offline-Zeit zu verkürzen, werden die eingangs in diesem Abschnitt genannten zwei zusätzlichen Phasen – Push-Phase und Pull-Phase – eingesetzt. Während dieser beiden Phasen läuft die virtuelle Maschine weiter, wobei parallel zur VM-Ausführung ein Teil der Speicherseiten transferiert wird. Lediglich ein kleiner Rest des Speichers (der während der Laufzeit nicht kopiert werden kann) wird in der Stop-and-Copy-Phase kopiert, so dass die Offline-Zeit reduziert wird. Verfahren, bei denen die VM während des Migrationsvorgangs weiterläuft, werden *Live-Migrationsverfahren* genannt.

Bei der Live-Migration können drei Ansätze verfolgt werden: *Pre-Copy* und *Post-Copy* bzw. *Bedarfsmigration* [6][7] sowie die Kombination von Pre-Copy- und Post-Copy-Ansätzen (hybrider Ansatz [8]). Tabelle 1 zeigt, welche Migrationsansätze aus welchen der drei genannten Phasen bestehen.

Bei dem *Pre-Copy*-Ansatz, der von Xen [6][10] unterstützt wird, besteht die Live-Migration aus der Push- und Stop-and-Copy-Phase. Bei dem *Post-Copy*-Ansatz, wie in [8] beschrieben, wird die VM nach dem Kopieren der zur Ausführung der VM notwendigen Zustandsinformationen in der Stop-and-Copy-Phase auf dem Zielhost gestartet und anschließend werden einzelne Speicherseiten beim Zugriff auf diese von dem Quellhost geholt. VMware setzt bei der VM-Live-Migration sowohl Push-, als auch Pull-Phasen ein und entspricht damit einer Kombination aus Pre-Copy- und Post-Copy-Ansatz, was als hybrider Ansatz bezeichnet wird [11].

Tabelle 1: Migrationsansätze.

Phase	Live Migration			Stop-and-Copy-Migration (klassischer Ansatz)
	Pre-Copy	Post-Copy (Bedarfsmigration)	Hybrider Ansatz	
Push	ja	nein	ja	nein
Stop-and-Copy	ja	ja	ja	ja
Pull	nein	ja	ja	nein

## 1.2. Anwendungen der Live-Migration

Derzeit werden Live-Migrationstechnologien vor allem für Wartungszwecke in Rechenzentren [12] eingesetzt: Wenn z. B. einige Komponenten eines physikalischen Hosts (oder komplette Host-Hardware) repariert, ausgetauscht oder Updates der Virtualisierungssoftware erforderlich werden, sind die virtuellen Maschinen von diesem Host zu migrieren. Es gibt aber auch andere Einsatzszenarien: Wenn die Auslastung einer virtuellen Maschine steigt, so dass die Host-Maschine überlastet wird, kann die Migration dieser VM zu einer besseren Verteilung der Last führen, die eine signifikante Performance-Steigerung bringen kann [13]. Live-Migration wird weiterhin u. a. verwendet, um durch Trennung der Hard- und Software den z. T. sehr dynamischen Ressourcen-Bedarf der Nutzer, vor allem in Cloud-Umgebungen, zu genügen. Die Nutzer können zwischen virtuellen Maschinen mit unterschiedlichen Rechenleistungen (und Kosten) wählen, mit der Option, bei Bedarf in Zukunft auf eine andere Hardware zu wechseln. Die beschriebenen Migrationsszenarien werden aktuell typischerweise durch den Administrator manuell durchgeführt, wobei die Automatisierung dieses Prozesses eine schnellere Reaktion auf Änderungen der Anforderungen ermöglichen würde und somit zukünftig vorgezogen werden sollte. Neben den genannten Einsatzszenarien erlaubt die automatische Live-Migration weitere Einsatzgebiete, von denen einige im folgenden skizziert werden.

## 2. Automatische Live-Migration

Für die automatische Live-Migration wird durch das Virtualisierungssystem eigenständig berechnet, wann und auf welchen Zielhost eine virtuelle Maschine migriert werden soll. Die Live-Migration wird anschließend selbstständig angestoßen. Diese Technologie erweitert das Anwendungsspektrum von Live-Migrationen durch eine Vielzahl von Möglichkeiten, wie z. B. automatisierte Migration bei Gefahr für die IT-Infrastruktur (sowohl bei der Gefahr eines Hardware-Defekts als auch bei Erkennung von Eindringlingen durch Intrusion-Detection-Mechanismen) oder die Migration auf energieeffizientere Zielsysteme für Kostensenkungsmaßnahmen. Eine Art automatische Live-Migration wird von VMware eingesetzt, die für die „Anpassung“ der IT-Infrastruktur an die dynamisch verändernde Nachfrage [5] implementiert worden ist. Die Migrationsentscheidung zur Berechnung von Zielhosts der automatischen Live-Migration beinhaltet in diesem Fall die Auslastung der Quell- und Zielhosts und die zur Verfügung stehenden Ressourcen (Speicher und CPU) auf dem Zielhost.

Wie in [14] dargestellt, gibt es eine Vielzahl weiterer Kriterien, die sowohl für bessere Performance als auch für bessere Ressourcenauslastung der zu migrierenden virtuellen Maschinen zu betrachten sind. Diese Migrationskriterien sind i. A. anwendungsspezifisch. Bei der Verwendung der automatischen Live-Migration für die Verbesserung der Energieeffizienz von Rechenzentren, sollte die Berechnung der neuen Zielhosts aktuelle Informationen über die Kühlungsart der in Frage kommenden Zielhosts (z. B. Kühlung durch Lüftung oder Kältetechnik) oder die Vermeidung von örtlich begrenzten heißen Bereichen im Serverraum (sog. thermischen Hotspots) berücksichtigen.

Im Falle von komplexen verteilten Anwendungen kann die Menge der Kommunikation der Teilprozesse untereinander in die Betrachtung einbezogen werden, um stark kommunizierende Prozesse unterschiedlicher virtueller Maschinen in eine örtliche Nähe zu migrieren. In [14] wird ein Clustering-Algorithmus als Grundlage einer Migrationsentscheidung vorgeschlagen, um diese zusätzlichen Beziehungen der involvierten Hosts in die Berechnung der Zielhosts einzubeziehen.

Wie bereits dargestellt, gibt es je nach Anwendungsfall unterschiedliche Migrationskriterien, die für eine automatische Live-Migration betrachtet werden müssen. Für nahezu alle Anwendungen gilt jedoch, dass die Migrationszeit bzw. die Offline-Zeit während der automatischen Live-Migration minimal gehalten werden sollte.

Aus diesem Grund werden in dem vorliegenden Beitrag Messungen präsentiert, die den Einfluss unterschiedlicher CPU- und Speicherauslastungsfälle auf die Performance des VM-Migrationsprozesses in einer Xen-basierten Umgebung darstellen. Ein besonderes Augenmerk wird auf die Offline-Zeiten der VM während der Migration gelegt. Die Ergebnisse dieser Messungen werden später in diesem Beitrag diskutiert, um für zukünftige automatische Live-Migrationen Lastszenarien zu vermeiden, in denen Live-Migrations- und Offline-Zeiten besonders hoch werden können.

### 3. Performance der Live-Migration

Für die im Folgenden präsentierten Performanceanalysen wurden zwei Server mit Xen-Hypervisor in der Version 4.01 und zwei virtuelle Linux-Maschinen verwendet. Messungen wurden einmal mit aktuellen Rechnersystemen durchgeführt und anschließend mit etwas älteren Rechnersystemen wiederholt, um den Effekt starker Ressourcenknappheit zu simulieren.

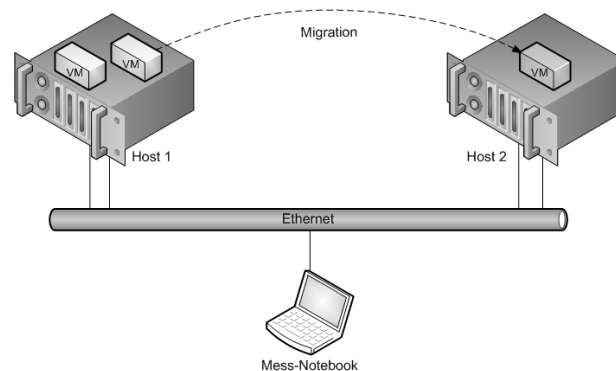


Abbildung 1: Testaufbau zum Messen des Datendurchsatzes.

Für die ersten Messungen wurden zwei Hosts mit Intel i5-660 Prozessoren (2 Cores mit je 2 Threads und 3,33 GHz Taktfrequenz), 4 GB RAM und 1-Gbit-Ethernet-Anschlüssen eingesetzt. Für Vergleichsmessungen auf den ressourcenarmen Systemen waren zwei Server mit je zwei Pentium III-Prozessoren mit 1,4 GHz, 1 GB RAM und zwei 1 Gbit-Ethernet-Anschlüssen im Einsatz. Es wurden auf beiden Systemen 128 MB RAM für jede virtuelle Maschine reserviert. Der schematische Testaufbau ist Abbildung 1 zu entnehmen. Beide Systeme wurden unabhängig voneinander den Tests unterzogen.

Zur Messung der Migrationsdauer und Offline-Zeiten wurde die Software Ixia IxChariot [15] auf dem Mess-Notebook eingesetzt. Es wurde ein UDP-basierter Dienst auf der migrierten VM gestartet und vom Messnotebook in kurzen Zeitabständen der Datendurchsatz (Throughput) beim Zugriff auf diesen Dienst auf Anwendungsebene gemessen (Abbildung 2).

Beim Betrachten des Durchsatzes über die Zeit können die Migrationsphasen eindeutig identifiziert werden. Abbildung 2 zeigt dies anhand einer Beispielmessung. Anhand eines ersten Leistungseinbruchs ist zu erkennen, wann die Migration beginnt (und damit ihre Push-Phase, in der die Speicherseiten in mehreren Iterationen kopiert werden, während die VM weiterläuft und damit erreichbar bleibt). Der Leistungseinbruch ist durch die parallele Last begründet, die der Kopiervorgang der Daten erzeugt (u. a. Last auf Speicherbus und Netzwerk).

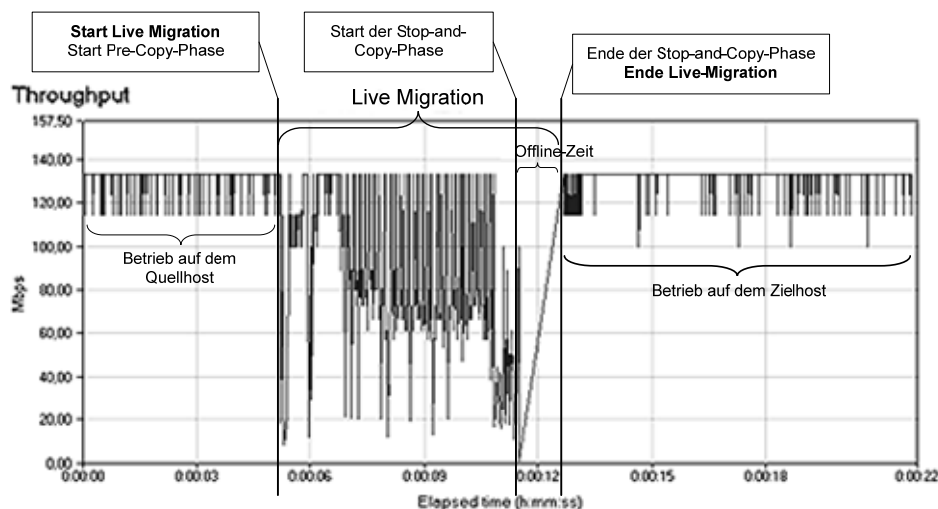


Abbildung 2: Durchsatz zwischen Messnotebook und virtueller Maschine beim Zugriff auf einen UDP-basierten Dienst der VM während der Live-Migration zwischen zwei physikalischen Hosts.

Ein weiterer Einbruch (ab dem gar keine Daten mehr übertragen werden) signalisiert, wann die Offline-Zeit beginnt und endet. Die Offline-Zeit entspricht der Stop-and-Copy-Phase, weil die virtuelle Maschine zu dieser Zeit angehalten wird und somit nicht mehr erreichbar ist, was den komplett fehlenden Datendurchsatz erklärt. Auf diese Weise können Migrationsdauer und VM-Offline-Zeiten ausgewertet werden. Die gering unterschiedlichen Durchsätze beim Zugriff auf dem UDP-basierten Dienst vor und nach der Migration der VM auf Quell- und Zielhost sind durch leichte Unterschiede der Hintergrundlast auf den beiden physikalischen Hosts zu erklären.

Um die Lastabhängigkeit der Migrationszeiten zu bestimmen, wurde die Prozessorlast mit Hilfe eines selbst entwickelten Lastgenerators variiert. Das Tool verändert Zeitphasen mit einer hohen CPU-Last (es wurden Fließkommarechnungen durchgeführt) und geringer Last (idle) so, dass die gesamte CPU-Last dem gewünschten prozentualen Wert entspricht. Die CPU-Belastung wurde mit Hilfe des Xen-Tools `xm top` innerhalb der Domain-0 kontrolliert.

Für die Variation der Anzahl der geänderten Seiten während der Migration der VM wurde das Last-Generator-Tool um die Möglichkeit erweitert, während der CPU-Auslastungsphasen auch eine angegebene Anzahl von Speicherseiten zu ändern.

Mit der vorgestellten Technik wurden folgende Messungen durchgeführt, deren Ergebnisse in den folgenden Abschnitten präsentiert werden:

1. Migrationsdauer und VM-Offline-Zeiten in Abhängigkeit von der CPU-Auslastung der migrierten virtuellen Maschine und in Abhängigkeit einer variablen Hintergrundlast auf dem physikalischen Host.
2. Migrationsdauer und VM-Offline-Zeiten in Abhängigkeit von der Speichergröße der VM und der Anzahl der von der VM während der Migration geänderten Speicherseiten.

### **3.1. Messung 1: Migrationsdauer und Offline-Zeiten in Abhängigkeit der CPU-Last**

Die erste Messung dient der Ermittlung der Abhängigkeit zwischen CPU-Last einer zu migrierenden virtuellen Maschine und der Migrationsdauer sowie Offline-Zeiten während der VM-Live-Migration.

Wie in der Abbildung 3 gezeigt, bleibt die Migrationsdauer auf dem leistungsstarken System fast konstant (linkes Chart) und hat nur einen Anstieg, wenn die Last der zu migrierenden VM 10% übersteigt. Dieses Verhalten ist zu erwarten, da aktuelle Hypervisor-Systeme die einzelnen virtuellen Maschinen so stark voneinander isolieren sollten, dass die VMs nur marginale Effekte aufeinander haben. Allerdings zeigen Messungen auf dem leistungsschwächeren Testsystem, dass diese Isolation im Fall knapper CPU- und Speicherressourcen nicht gewährleistet werden kann: Die Migrationszeit beginnt anzusteigen, wenn die CPU-Last in der zu migrierenden VM etwa 60% übersteigt, wie im rechten Chart der Abbildung 3 zu erkennen ist.

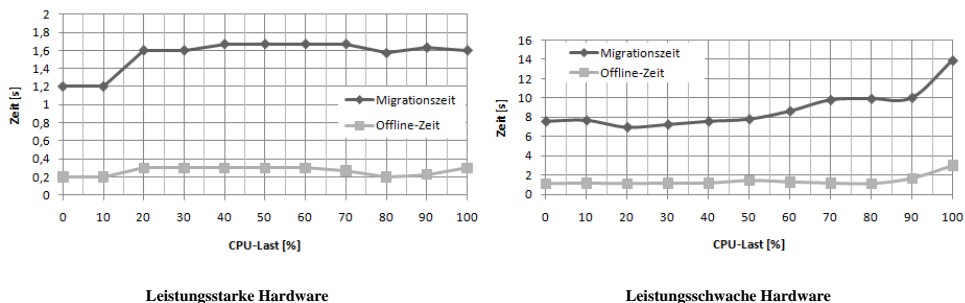


Abbildung 3: Migrationsdauer und Offline-Zeiten während der Live-Migration einer virtuellen Maschine bei gleichzeitiger CPU-Lasterhöhung innerhalb der migrierten VM.

Die VM-Offline-Zeiten bei der Live-Migration sind auf beiden Testsystemen etwa konstant. Auf dem älteren Pentium III-basierten System steigen die Offline-Zeiten ca. um den Faktor zwei auf ca. 3 Sekunden, wenn die virtuelle CPU der Maschine eine Auslastung von über 90% hat.

In den Messungen, die Abbildung 3 zugrunde lagen, wurden die physikalischen Hosts nur durch die migrierte VM belastet, d. h. die Hintergrundlast wurde minimiert. Die in Abbildung 3 erkennbaren Ergebnisse konnten auch nachgewiesen werden, wenn die Messungen mit einer hohen Hintergrundlast auf dem Quellhost wiederholt wurden.

Die Hintergrundlast wurde durch eine zweite virtuelle Maschine erzeugt, bei der eine 100%-Last der CPU durch den Lastgenerator produziert wurde. Die Ergebnisse dieser Messungen sind in Abbildung 4 zu sehen. Das Verhalten des leistungsstarken Systems mit den i5-Core-Prozessoren ist der Abbildung 3 ähnlich: Die Migrationszeiten und Offline-Zeiten der migrierten VM bleiben stabil und unabhängig von der eigenen CPU-Last und der Hintergrundlast in der zweiten VM.

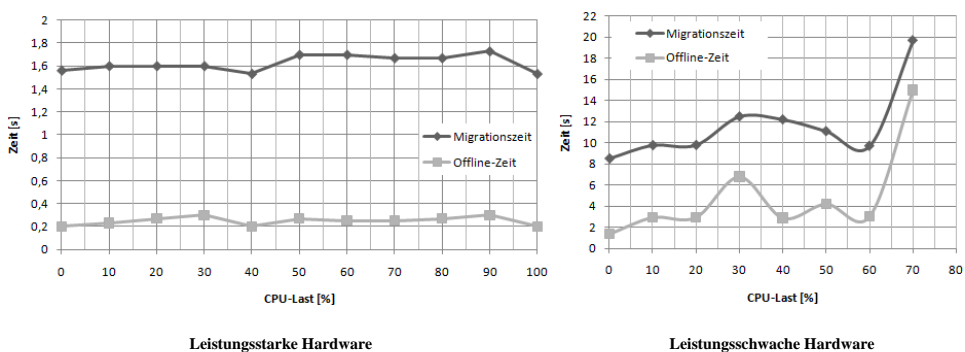


Abbildung 4: Migrationsdauer und Offline-Zeiten einer VM mit variabler Last in der migrierten virtuellen Maschine bei konstant hoher CPU-Hintergrundlast auf dem Quellhost



Auf dem leistungsschwachen System wurden die Offline-Zeiten der VM dagegen wesentlich größer und dauerten bis zu 15 Sekunden, wenn die CPU-Last der migrierten VM auf über 60% anstieg. Bei über 70% Last wurden die Offline-Zeiten so groß, dass die Verbindung zwischen Messsystem und virtueller Maschine aufgrund eines Timeouts abbrach. Messungen der Offline-Zeiten für Lasten oberhalb der 70% waren deshalb nicht möglich.

Aus beiden Messungen ist also erkennbar, dass auf dem leistungsschwachen System sowohl bei niedriger als auch bei hoher Hintergrundlast ab einer Auslastung der virtuellen CPU der zu migrierenden Maschine von über 60% eine Zunahme der Migrationszeiten zu erwarten sind. Im Falle von hoher Hintergrundlast kann diese Zunahme so deutlich sein, dass es zu Verbindungsabbrüchen zwischen externen Rechnern und der migrierten VM kommen kann. Diese Tatsache könnte insbesondere ein Problem bei alter Server-Hardware sein.

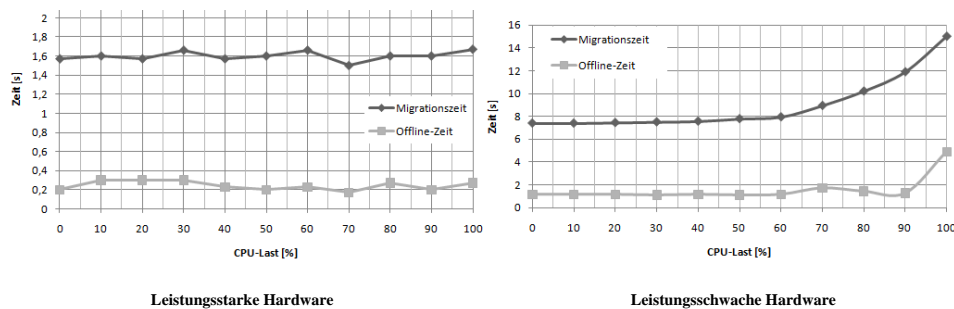


Abbildung 5: Migrationsdauer und Offline-Zeiten einer VM in Abhängigkeit von CPU-Hintergrundlast des Quellhosts durch variable Lasterzeugung in einer weiteren virtuellen Maschine

In einem weiteren Versuch (Abbildung 5) wurde die CPU-Last einer laufenden virtuellen Maschine während der Migration einer anderen VM ohne Last variiert. Das leistungsstarke Testsystem ließ dabei stabile Migrations- und Offline-Zeiten erkennen. Auf dem älteren System erhöhte sich die Dauer der Live-Migration ab ca. 60% der CPU-Last der nicht migrierten VM, wobei die Offline-Zeit konstant blieb und erst ab ca. 90% anstieg.

### 3.2. Messung 2: Migrations- und Offline-Zeiten in Abhängigkeit der Hauptspeicherlast

In diesem Abschnitt wird präsentiert, welchen Einfluss die Hauptspeichergröße virtueller Maschinen und die Anzahl der innerhalb der VM geänderten Seiten auf die VM-Migrationsdauer und die Offline-Zeiten haben. Die unten dargestellten Messungen beziehen sich auf das leistungsschwächere Sekundärsystem, da dieses System repräsentativ zeigt, welches Verhalten ein System mit Leistungsengpässen hat.

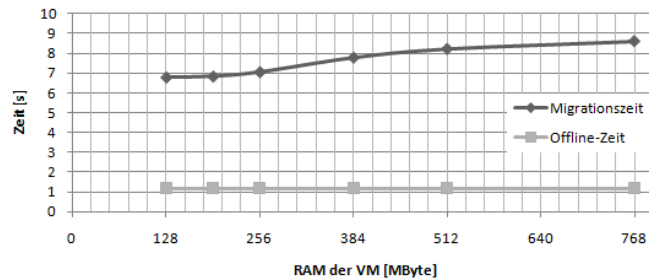


Abbildung 6: Migrationsdauer und Offline-Zeit während der Live-Migration in Abhängigkeit von variierender VM-Speichergröße.

Abbildung 6 demonstriert die Änderungen der Migrationsdauer und Offline-Zeiten einer VM in Abhängigkeit von der variierenden Speichergröße dieser Maschine. Es ist erkennbar, dass die Migrationsdauer in einer annähernd linearen Abhängigkeit von der Speicherkapazität der VM steigt. Dies ist ein vorhersehbares Verhalten, da die höhere Anzahl der Speicherseiten während der Push-Phase der Live-Migration kopiert werden muss.

Es ist auch gut erkennbar, dass die Offline-Zeiten, die der Dauer der Stop-and-Copy-Phase bei der Migration entsprechen, nahezu konstant bei allen Messungen sind. Da die laufenden Anwendungen innerhalb der VM während dieser Testmessung immer eine ähnliche Anzahl von Speicherseiten geändert haben, wird vom Xen-Hypervisor auch immer eine ähnliche Anzahl von geänderten Speicherseiten während der Stop-and-Copy-Phase zum neuen Host gesendet.

Wenn eine Anwendung viele Seitenänderungen während der VM-Migration verursacht, sollten sich die Offline-Zeiten dieser VM entsprechend erhöhen. Diese Tatsache ist in Abbildung 7 erkennbar. Für diesen Test wurde eine Lastapplikation innerhalb der VM ausgeführt, die stetig neue Speicherseiten modifiziert und so zu einer Verdopplung der Dauer der Stop-and-Copy-Phase führt.

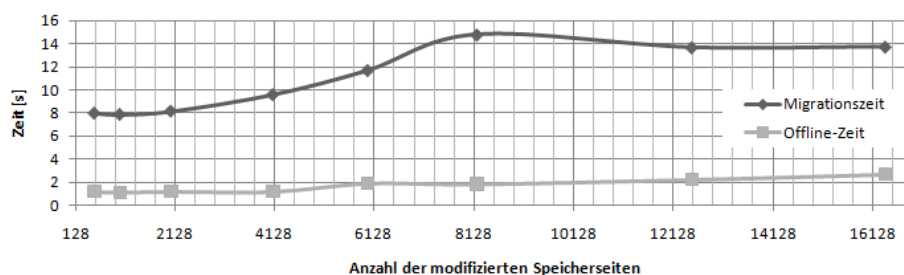


Abbildung 7: Migrations- und Offline-Zeiten bei Variation der modifizierten Speicherseiten in der VM.

### 3.3. Ergebnisse

Während die präsentierten Messungen Unterschiede im Detail zeigen, können die folgenden wesentlichen Schlussfolgerungen für unsere Xen-basierte Testumgebung gezogen werden:

- Die Dauer der Live-Migration einer virtuellen Maschine bleibt mit steigender Last innerhalb der VM stabil und ist somit lastunabhängig.
- Die Migrationszeit einer virtuellen Maschine auf einen anderen physikalischen Host steigt stark an, wenn mehrere virtuelle Maschinen derart hohe Last erzeugen, dass die physikalischen Ressourcen (CPU und Arbeitsspeicher) des Quellhosts ihre Kapazitätsgrenzen erreichen.
- Offline-Zeiten während der Migration sind relativ unabhängig von der CPU-Last.
- Die VM-Migrationsdauer hängt von der Speichergröße der virtuellen Maschine in linearer Weise ab. Die Offline-Zeit während der Migration ist unabhängig von der Größe des VM-Speichers.
- Viele Speichermodifikationen während der Migration führen zur Erhöhung der Migrationsdauer und der Offline-Zeit gleichzeitig.

## 4. Diskussion

Diese Ergebnisse führen zu Hinweisen, die beachtet werden sollten, um hohe Migrations- und Offline-Zeiten von Live-Migrationen zu vermeiden. Diese Hinweise können von Systemadministratoren für manuelle Live-Migrationsszenarien eingesetzt werden, um die Performance zu verbessern, sind aber vor allem im Umfeld der automatischen Live-Migration von Interesse. Gerade in automatischen Live-Migrationsszenarien können je nach Migrationsentscheidung jederzeit Migrationen auftreten, so dass ein störungsfreier Betrieb nur gewährleistet werden kann, wenn die Migrations- und insbesondere die Offline-Zeit der migrierten VM möglichst klein ist. In Szenarien mit automatischer Live-Migration sollten entsprechend die folgenden Hinweise bei der Migrationsentscheidung berücksichtigt werden:

- Auf aktuellen Serversystemen sind Migrations- und Offline-Zeiten unabhängig von der Last virtueller Maschinen. Bei der Verwendung älterer Serversysteme sollte die Migration zweier stark belasteter VMs auf den gleichen physikalischen Host vermieden werden, um hohe Migrationszeiten zu verhindern.

- Die VM-Speichergröße beeinflusst die Migrationsdauer, aber nicht die Offline-Zeiten während der Migration. Die Offline-Zeit wird nur durch Workloads von Anwendungen innerhalb der VM beeinflusst, die viele Speicherseiten während der Migration modifizieren. Zur Reduktion der Offline-Zeiten von solchen Anwendungen sollte die betroffene VM nur zum Zeitpunkt mit minimalen Speicheränderungen migriert werden (z. B. nachts, wenn Änderungen wegen weniger Benutzer-Aktivität reduziert werden). Für dieses Migrationskriterium muss die Menge der modifizierten Speicherseiten pro VM stetig geloggt werden. Bei vielen Seitenmodifikationen pro Zeiteinheit sollte eine Migration vermieden werden. Da die Funktionalität der Zählung modifizierter Seiten für die Realisierung der Push-Phase einer Live-Migration im Hypervisor implementiert sein muss und in aktuellen Betriebssystemen auch aus anderen Gründen realisiert ist, kann eine Nutzung der vorliegenden Daten als Live-Migrationskriterium in vielen Fällen recht einfach erfolgen.

## 5. Zusammenfassung

In diesem Beitrag wurden Messungen präsentiert, die die Migrations- und Offline-Zeiten virtueller Maschinen in Abhängigkeit von deren CPU-Last und Speichernutzung darstellten. Aus den Ergebnissen konnten zwei Kriterien formuliert werden, die beachtet werden sollten, um die Dauer von Live-Migrationen und deren Offline-Zeiten gering zu halten. Die Kriterien basieren auf Messungen in einer Xen-basierten Umgebung und sind somit im ersten Schritt nur für Xen-Virtualisierungssysteme validiert.

Da Xen als Migrationsmethode den Pre-Copy-Ansatz verwendet, können Rückschlüsse auf Systeme mit anderen Migrationsmethoden (wie VMware) nicht direkt erfolgen. Es ist Thema von zukünftigen Untersuchungen, auch die Übertragbarkeit der Kriterien auf andere Virtualisierungssysteme und Hardwareplattformen zu validieren.

## 6. Danksagung

Wir möchten dem Ministerium für Wirtschaft, Arbeit und Tourismus Mecklenburg-Vorpommern für seine finanzielle Unterstützung des Projektes danken. Wir danken auch unserem Projektpartner INR Informationssysteme Rostock für die freundliche und fruchtbare Zusammenarbeit im Verbundprojekt.

## Literaturverzeichnis

- [1] Jeff Daniels. Server virtualization architecture and implementation. Crossroads, Vol. 16, 2009.
- [2] Mark Huff. Why Virtualize? Whitepaper, January 2009. [http://www.technologent.com/Documents/WhyVirtualizev\\_Jan09.pdf](http://www.technologent.com/Documents/WhyVirtualizev_Jan09.pdf)
- [3] Robert Rose. Survey of System Virtualization Techniques, March 2004. <http://citeseer.ist.psu.edu/720518.html>

- [4] Matthias Richly and Christopher Schuster. Dependable Systems 2010. Virtualization Fault Tolerance. Project Report, 2010.
- [5] VMware Inc. How VMware Virtualization Right-sizes IT Infrastructure to Reduce Power Consumption, 2008.
- [6] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live Migration of Virtual Machines. In *NSDI'05 Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, page 14. ACM, 2005.
- [7] Tzi-cker Chiueh and Susanta Nanda. A Survey on Virtualization Technologies, 2005.
- [8] Michael Hines, Umesh Deshpande, and Kartik Gopalan. Post-Copy Live Migration of Virtual Machines. In: *ACM SIGOPS Operating Systems Review*; Vol. 43; iss 3; pg. 14-26; 2009.
- [9] VMware Inc. Resource Management with VMware DRS, 2006.
- [10] Xen hypervisor, the powerful open source industry standard for virtualization. <http://www.xen.org/>
- [11] Michael Nelson, Beng-Hong Lim, and Greg Hutchins. Fast Transparent Migration for Virtual Machines. In: *Proceedings of the 2005 USENIX Annual Technical Conference*. pg. 391-394. 2005.
- [12] Ingard Mevåg. Towards Automatic Management and Live Migration of Virtual Machines. Master's Thesis. University of Oslo, 2007.
- [13] Sandra Gittlen. Planning for virtualization? Beware of server overload. *Computerworld Magazine*. Feb. 2010.
- [14] Daniel Versick, Djamshid Tavangarian. Reducing Energy Consumption by Load Aggregation with an Optimized Dynamic Live Migration of Virtual Machines. In: *Proceedings of the 2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. pg. 164-170, IEEE 2010.
- [15] IxChariot - Test Device and System Performance under Real-World Load Conditions. <http://www.ixchariot.com/products/datasheets/ixchariot.html>.